# VQ-TEGAN: Data Augmentation with Text Embeddings for Few-shot Learning

Anonymous ACL submission

## Abstract

Data augmentation is crucial for the fine-tuning of pre-trained models and the optimization of limited data utilization, particularly within the realm of few-shot learning. Traditionally, these techniques have been applied at the word and sentence levels, with little research conducted within the embedding space. This paper introduces **VQ-TEGAN**, a novel data augmentation approach designed to generate embeddings specifically for a few-shot learning. VQ-TEGAN generates embeddings that augment the few-shot dataset by training directly within the PLMs' word embedding, employing a customized loss function. Empirical valildation on GLUE benchmark datasets demonstrates that VQ-TEGAN markedly improves text classification performance. Additionally, we investigate the application of VQ-TEGAN with RoBERTa-large and BERT-large, offering insight for further application.

## 1 Introduction

Text classification is a crucial task in natural language processing (NLP) (Kowsari et al., 2019). Although fine-tuning pre-trained language models (PLMs) on large datasets is highly effective, performance declines with smaller training data sizes (Gao et al., 2020; Longpre et al., 2020). This is due to the lack of diverse examples. Data augmentation has emerged as a solution to improve model performance with limited data, applicable in various fields such as healthcare (Eaton-Rosen et al., 2018; Ker et al., 2017), finance (Fons et al., 2020; El-Laham and Vyetrenko, 2022), and computer vision (Zhang et al., 2017; Chen et al., 2020b).

In NLP, data augmentation is often performed through word-level manipulation (*e.g.*, EDA (Wei and Zou, 2019) and AEDA (Karimi et al., 2021)). Recent advances include sentence-level interpolation methods like MixText (Zhang et al., 2022) and Treemix (Zhang et al., 2022; Chen et al., 2020a).

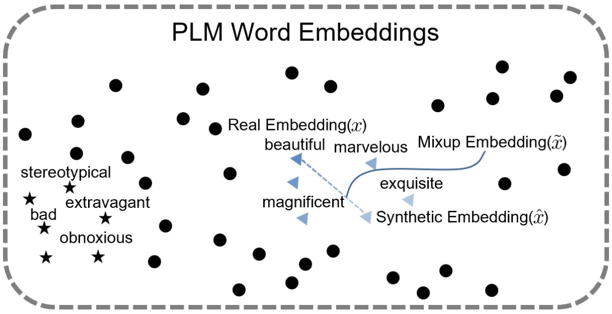



Figure 1: Graphical abstract of VQ-TEGAN. The primary aim of VQ-TEGAN is to produce synthetic embeddings that closely approximate the original real embeddings. Subsequently, the synthetic embedding is mixed with the real embedding to formulate a mixup embedding, which resides within a space comparable to that of other synonymous embeddings.

In addition, language-model-based augmentations such as LAMBADA (Anaby-Tavor et al., 2020), BF-Translation (Body et al., 2021), BART ProtAugment (Dopierre et al., 2021), and SSMBA (Ng et al., 2020) have been developed. While LAMBADA and BART ProtAugment require separate fine-tuning for data augmentation, SSMBA and BF-Translation do not, but they demand significant storage space and time due to the need for large language models or the Google Translation API.

Before training a language model, sentences are tokenized and converted to embeddings, which are used as direct input (Mikolov et al., 2013). Some works have applied data augmentation at the embedding level. For example, Wang and Yang (2015) used semantic and lexical embeddings from Word2Vec (Mikolov et al., 2013) to replace original words with $k$-nearest neighbor vectors. TreeMixup (Guo et al., 2019) applies linear interpolation to word and sentence embeddings, pioneering this technique in NLP tasks. TACLR (Jia et al., 2023) combines TreeMixup and EDA for contrastive learning. Recent studies show promising results using models that generate synthetic sentence embeddings similar to real sentences (Onan, 2023; Jian et al., 2022). These methods effectively enhance text embeddings to supplement insufficient data.

This research proposes **V**ector-**Q**uantized **T**ext **E**mbedding **G**enerative **A**dversarial **N**etworks (**VQ-TEGAN**). VQ-TEGAN leverages the capabilities of Vector Quantized Generative Adversarial Network (VQ-GAN) (Esser et al., 2021) to generate text embeddings optimized for the semantic representation provided by word embeddings in PLMs (*e.g.*, RoBERTa-large (Liu et al., 2019) and BERT-large (Devlin et al., 2018)). VQ-TEGAN is based on the understanding that the word embeddings of PLMs can capture deep linguistic properties beyond simple syntactic structures. We hypothesize that synthetic embeddings generated by VQ-TEGAN can encapsulate complex features such as context and sentiment, crucial for few-shot learning tasks. Synthetic embeddings are employed in PLM training to provide new text examples that preserve semantic consistency and syntactic accuracy with the few-shot embedding data. This approach aligns with Brown et al. (2020), demonstrating that language models trained on extensive datasets can leverage prior knowledge to perform tasks with limited examples.

Our contributions can be summarized as follows:

- We propose a novel data augmentation model, VQ-TEGAN, for generating synthetic embeddings located in a similar space as real embeddings as illustrated in Figure 1.
- VQ-TEGAN is a lightweight model for data augmentation, allowing easy application and minimal storage requirements.
- We introduce a novel loss function suitable for NLP embeddings to train VQ-TEGAN.
- Experimental results indicate that VQ-TEGAN outperforms benchmarks in few-shot learning.
- The adequacy of the generated embeddings is confirmed by analyzing their meaning using cosine similarity to the word embeddings in PLMs.

## 2 Related Work

### 2.1 Generative Model

The evolution of generative models has been led by the advances of autoencoders (Ranzato et al., 2007). Variational Autoencoders (VAE) (Kingma and Welling, 2013) use neural networks to encode input data into a lower-dimensional latent space and decode it back, optimizing the lower bound on the likelihood of the data. This enables tasks such as data generation and feature extraction. Generative Adversarial Networks (GAN) (Goodfellow et al., 2014) employ two neural networks, a generator and a discriminator, training them simultaneously in a competitive setting to generate data samples that are indistinguishable from real data. Wasserstein GAN (WGAN) (Arjovsky et al., 2017) improves on traditional GANs by using a Wasserstein distance metric for the loss function, improving training stability and addressing mode collapse, resulting in higher-quality generated samples. Conditional WGAN (cWGAN) (Yu et al., 2019) extends WGAN by incorporating conditional variables, allowing the generation of samples conditioned on specific attributes and enhancing the model's ability to generate more targeted and diverse data samples. Vector Quantized Variational Autoencoders (VQ-VAE) (Van Den Oord et al., 2017) and VQ-GAN employ discrete latent representations through vector quantization. VQ-VAE improves its ability to handle complex data distributions compared to standard VAEs by learning a finite set of embeddings. VQ-GAN combines the VQ-VAE method with a discriminator to differentiate between real and fake data more effectively by learning a codebook.

In the realm of NLP, autoencoders are frequently combined to generate data in an embedding space (Malandrakis et al., 2019; Piedboeuf and Langlais, 2022). This study leverages the VQ-GAN method to generate synthetic embeddings. Additionally, we analyze the semantic content of the synthetic embeddings produced by VQ-TEGAN and compare it with the embeddings created by mixup and the original text embedding data.

### 2.2 Text Augmentation

Text augmentation aims to improve model performance when data is insufficient. Early work includes EDA (Wei and Zou, 2019) and AEDA (Karimi et al., 2021). EDA employs four straightforward data augmentation techniques: random swap, random insertion, random deletion, and synonym replacement. Similarly, AEDA operates by randomly inserting punctuation marks. TreeMix (Zhang et al., 2022) utilizes constituency parsing trees to decompose sentences into component substructures, which are then recombined using the mixup data augmentation method to generate new sentences.

Instead of reorganizing words or sentences, another approach involves generating new text data using LLMs for data augmentation (Anaby-Tavor et al., 2020; Body et al., 2021; Dopierre et al., 2021; Ng et al., 2020). LAMBADA (Anaby-Tavor et al., 2020) fine-tunes a GPT model (Radford et al., 2019) on a small dataset and then augments it with the given label. BF-Translation (Body et al., 2021) uses the Google Translate API, with German as an intermediate language, to back-translate text for sentiment analysis. ProtAugment (Dopierre et al., 2021) combines paraphrases generated from the BART model with sentences produced through traditional back-translation, improving intent detection models via unsupervised meta-learning. This method utilizes paraphrasing-based data augmentation. SSMBA (Ng et al., 2020) is a word-level data augmentation technique that employs a corruption function to mask specific tokens in a sentence and replace them with new tokens using a BERT model.

Furthermore, data augmentation in continuous embedding spaces, such as EmbedHalluc (Jian et al., 2022), has shown promising results. Specifically, graph-based methods (Onan, 2023) and contrastive learning (Jia et al., 2023) have been explored for text augmentation. Embedding Augmenter (Pellicer et al., 2023) is a technique that uses a word-changing algorithm with the GloVe model (Pennington et al., 2014) with 300 dimensions to find the most similar words.

This study investigates the use of synthetic embeddings for data augmentation, where embeddings are derived from synonyms and related words. In particular, the proposed VQ-TEGAN model offers the advantage of being relatively lightweight compared to larger language models.

### 2.3 Fine-tuning of Pre-trained Language Models

Numerous studies suggest using general models to address NLP problems (Kim, 2014; Huang et al., 2015; Kowsari et al., 2019). However, with the recent emergence of PLMs (*e.g.*, BERT and RoBERTa), there has been a surge in research on few-shot learning to leverage limited data with the help of PLMs (Gupta et al., 2020; Zhong et al., 2021; Chada and Natarajan, 2021; Ram et al., 2021). Some studies have applied data augmentation to NLP classification tasks to improve few-shot learning performance (Wei et al., 2021; Jian et al., 2022; Zhang et al., 2022; Jia et al., 2023). However, the approach of creating new synthetic word embeddings for each word in a sentence, merging them, and using the resulting synthetic sentence embedding as training data for text classification has not yet been explored. In this context, we propose VQ-TEGAN, the first attempt to apply the VQ-GAN method to generate new synthetic text embeddings for fine-tuning PLMs.

## 3 Methods

### 3.1 Overview

This research aims to evaluate the effectiveness of VQ-TEGAN in few-shot learning compared to benchmarks by performing classification tasks in limited data environments. The complete process of fine-tuning the PLM is illustrated in Figure 2.

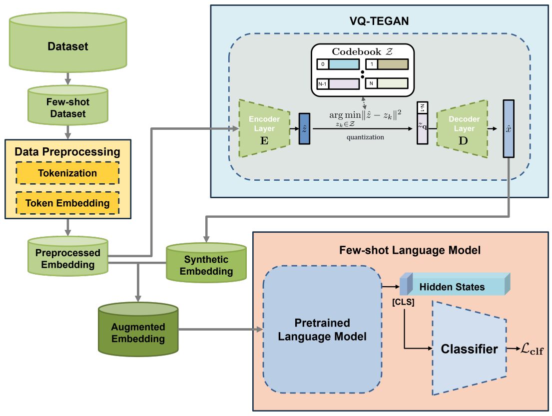



Figure 2: Few-shot learning process using VQ-TEGAN

To preserve the integrity and diversity of the dataset, non-duplicating samples are randomly selected from each class for each classification task in the training and validation sets, respectively. The conversion of few-shot datasets to real embeddings is achieved using the PLM's individual tokenizer and token embeddings, which are subsequently used to form preprocessed embeddings. The real embeddings of the training set are then utilized to create synthetic embeddings through the pre-trained VQ-TEGAN. The synthetic embeddings for each real embedding are subsequently mixed to form the final augmented embeddings. The final augmented dataset, which includes one synthetic data point corresponding to each real data point, is used for few-shot learning. This approach takes advantage of the diversity introduced by the augmented data, operating under the assumption that it will enhance the learning capacity of the model when dealing with a restricted dataset (Arthaud et al., 2021; Xie et al., 2020). Also, the freezing of word embeddings within the PLM during few-shot learning preserves the semantic integrity of the augmented dataset within the embeddings. This method proficiently transmits the intended semantics of the augmented dataset in few-shot learning contexts.

### 3.2 VQ-TEGAN

The architecture of a new generative model for text embedding data, VQ-TEGAN, is presented in detail in Figure 3. The primary objective is to train VQ-TEGAN directly within word embeddings in PLM to generate high-quality synthetic text embeddings. This approach has the advantage of leveraging PLM embeddings, eliminating the need for a separate training dataset. Furthermore, VQ-TEGAN allows the encapsulation of word embeddings with analogous attributes into quantized vectors, ensuring that the generated synthetic embeddings retain their distinct characteristics. The amount of training data depends on the number of word embeddings in PLMs. Note that RoBERTa-large and BERT-large have 50,265 and 30,522 embedding vectors, respectively. This approach has the advantage of utilizing embeddings of PLM, eliminating the need for a separate training dataset.

In VQ-VAE, a discrete-dimensional encoder output paired with an autoregressive decoder effectively solves the posterior collapse problem (Van Den Oord et al., 2017). VQ-TEGAN employs a similar structure to reconstruct the real embedding ($x$) as the synthetic embedding ($\hat{x}$) through the encoder $\mathbf{E}$ - decoder $\mathbf{D}$ structure illustrated in Figure 3. The input vector $x \in \mathbb{R}^{n_x}$, where $n_x$ is the dimensionality of the input embedding, is compressed by the encoder $\mathbf{E}$ into the latent vector $\hat{z} \in \mathbb{R}^{n_z}$, where $n_z$ is the dimensionality of the codebook vector.

The latent vector $\hat{z}$ is converted into one of the nearest codebook vectors, $z_{\mathbf{q}} \in \mathcal{Z}$, by finding the distance to the vectors in the predefined discrete codebook, where $\mathcal{Z} = \{z_k\}_{k=1}^{K} \subset \mathbb{R}^{n_z}$ and $K$ is the number of codebook vectors. Specifically, $\hat{z}$ is created from $x$ and then quantized by replacing $\hat{z}$ with the nearest codebook to obtain $z_{\mathbf{q}}$ such that:

$$z_{\mathbf{q}} = \mathbf{q}(\hat{z}) := \arg\min_{z_k \in \mathcal{Z}} \|\hat{z} - z_k\|^2 \in \mathbb{R}^{n_z} \quad (1)$$

where $\hat{z} = \mathbf{E}(x)$. The reconstruction $\hat{x} \approx x$ is given by:

$$\hat{x} = \mathbf{D}(z_{\mathbf{q}}) \quad (2)$$

Backpropagation is not differentiable due to the quantization operation in Eq. 1. However, the model and codebook can be learned end-to-end via a loss function using a straight-through gradient estimator (Bengio et al., 2013) that copies the gradient from the decoder to the encoder as follows:

$$\mathcal{L}_{\mathbf{VQ}}(\mathbf{E}, \mathbf{D}, \mathcal{Z}) = \|x - \hat{x}\| + 1 - \sigma(\hat{x}, x) + \\ \|\text{sg}[\mathbf{E}(x)] - z_{\mathbf{q}}\|^2 + \beta \times \|\text{sg}[z_{\mathbf{q}}] - \mathbf{E}(x)\|^2 \quad (3)$$

Note that $\|x - \hat{x}\|$ is a reconstruction loss ($\mathcal{L}_{\text{rec}}$); $1 - \sigma(\hat{x}, x)$ is the cosine loss ($\mathcal{L}_{\text{cos}}$) (Barz and Denzler, 2020) where $\sigma(\cdot)$ represents the cosine similarity operation; and $\|\text{sg}[z_{\mathbf{q}}] - \mathbf{E}(x)\|^2$ is the commitment loss (Van Den Oord et al., 2017) where sg$[\cdot]$ represents the stop-gradient operation.

To customize a learning approach for text embeddings, we modify the loss function commonly used in computer vision (Esser et al., 2021). Specifically, we replace the $L_2$ loss with the $L_1$ loss in $\mathcal{L}$rec, a technique known for its effectiveness in high-resolution image restoration tasks (Zhao et al., 2016; Wu et al., 2021; Liu et al., 2021). The importance of cosine similarity in semantic analysis is derived from the inherent nature of text data embedding (Rahutomo et al., 2012; Pellicer et al., 2023). $\mathcal{L}$cos is employed to ensure that the synthetic embedding $\hat{x}$ is generated in a space characterized by high cosine similarity to the real embedding $x$.

The discriminator of VQ-TEGAN, **Disc**, is responsible for distinguishing between real and fake embedding, resulting in a loss $\mathcal{L}_{\mathbf{Disc}}$ that follows the WGAN loss to efficiently train the generator (Arjovsky et al., 2017):

$$\mathcal{L}_{\text{GAN}}(\{\mathbf{E}, \mathbf{D}, \mathcal{Z}\}, \mathbf{Disc}) = \mathbf{Disc}(x) - \mathbf{Disc}(\hat{x}) \quad (4)$$

The complete objective to identify the optimal compression model $\mathcal{Q}^* = \{\mathbf{E}^*, \mathbf{D}^*, \mathcal{Z}^*\}$ can be expressed as follows:

$$\mathcal{Q}^* = \arg\min_{\mathbf{E}, \mathbf{D}, \mathcal{Z}} \max_{\mathbf{Disc}} \mathbb{E}_{x \sim p(x)} [\mathcal{L}_{\text{VQ}}(\mathbf{E}, \mathbf{D}, \mathcal{Z}) + \\ \mathcal{L}_{\text{GAN}}(\{\mathbf{E}, \mathbf{D}, \mathcal{Z}\}, \mathbf{Disc})] \quad (5)$$

VQ-TEGAN stands out for its scalability and memory efficiency in text embedding data augmentation, optimizing computational resources. The model's parameters remain almost constant despite an increase in codebooks, growing only slightly from 5.03M (19.22MB) for 1024 codebooks to 5.42M (20.72MB) for 4096 codebooks. This

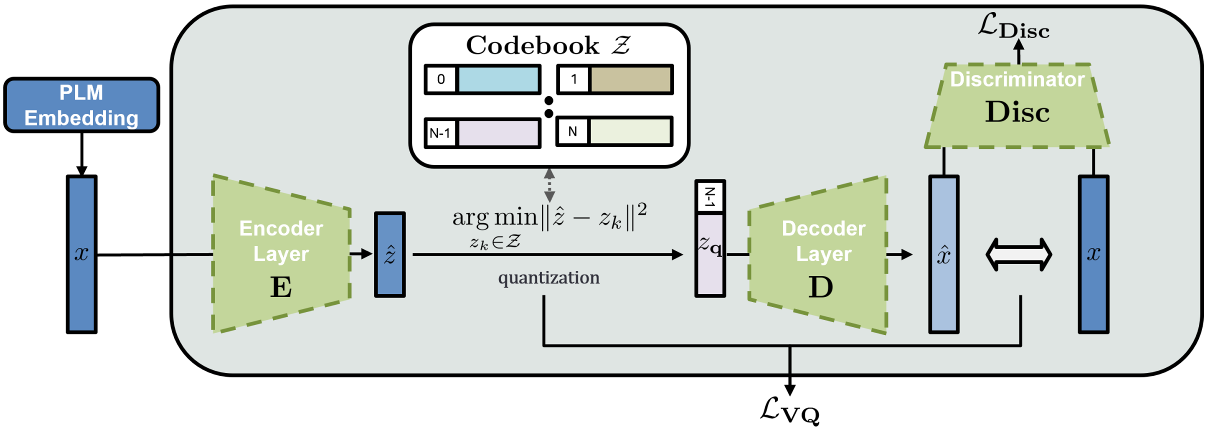



Figure 3: Model architectures of VQ-TEGAN

lightweight nature allows VQ-TEGAN to be deployed on various hardware, from high-end servers to resource-limited edge devices. Its compact design makes it ideal for scenarios that require robust text embedding augmentation without compromising performance. Training procedures are detailed in Appendix A.

### 3.3 Mixup Embedding

Mixup for word embedding, an application method devised by Guo et al. (2019), involves the linear interpolation of real and synthetic embeddings. We apply the mixup method as follows:

$$\tilde{x} = \lambda x + (1 - \lambda)\hat{x} \tag{6}$$

The mixup ratio $\lambda$ specifies the proportion of real embedding ($x$) in the mixed embedding. For instance, a $\lambda$ of 1.0 indicates that the mixed embedding $\tilde{x}$ is entirely composed of $x$, while a $\lambda$ of 0.4 produces a mixture of 40% of $x$ and 60% of $\hat{x}$. When $\lambda$ is 0.0, $\tilde{x}$ is composed of $\hat{x}$ only.

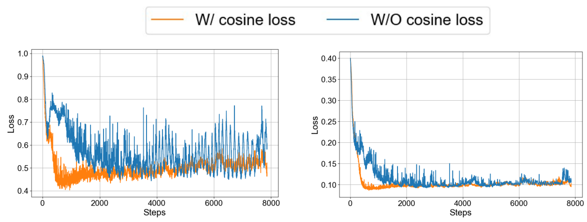
## 4 Results & Discussions

### 4.1 Dataset

The research employs nine classification tasks from the GLUE benchmark dataset (Wang et al., 2018). The GLUE benchmark encompasses diverse tasks, including grammatical acceptability (**CoLA**), sentiment analysis (**SST-2**), sentence semantic equivalence (**MRPC**), semantic similarity (**QQP**), logical inference (**MNLI-m**, **MNLI-mm**), validity of sentence answers to questions (**QNLI**), and logical entailment (**RTE**), pronoun resolution (**WNLI**).

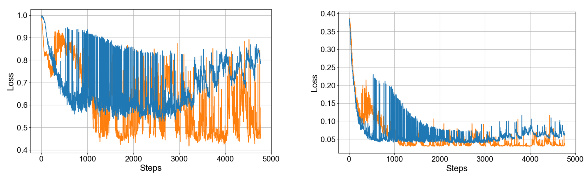
We randomly select 16 train and validation samples per class from the train and validation set of each task. The evaluations are based on the average results of five different seeds in the test set.

### 4.2 Impact of Consine Loss




(a) $\mathcal{L}_{\text{cos}}$ using RoBERTa-large word embedding

(b) $\mathcal{L}_{\text{rec}}$ using RoBERTa-large word embedding




(c) $\mathcal{L}_{\text{cos}}$ using BERT-large word embedding

(d) $\mathcal{L}_{\text{rec}}$ using BERT-large word embedding

Figure 4: $\mathcal{L}_{\text{cos}}$ and $\mathcal{L}_{\text{rec}}$ as a loss function with or without $\mathcal{L}_{\text{cos}}$ when training VQ-TEGAN

The analysis of Figure 4 underscores the importance of integrating the cosine loss term, $\mathcal{L}_{\text{cos}}$, within Eq. 3. The integration stabilizes and accelerates the convergence, thus enhancing model performance in similarity measures and improving the quality of the reconstructed data.

Figures 4a and 4c illustrate the effect of cosine loss on the cosine similarity between the real embedding $x$ and the synthetic embedding $\hat{x}$. The figures demonstrate that incorporating cosine loss in the generator's loss function, $\mathcal{L}_{\text{VQ}}$, leads to faster and more stable convergence (orange line) compared to the method without cosine loss (blue line) during VQ-TEGAN training.

Figures 4b and 4d illustrate the reconstruction loss, $\mathcal{L}_{\text{rec}}$, defined as the $L_1$ loss that quantifies the difference between real and synthetic embeddings. Incorporation of cosine loss yields lower and more stable $L_1$ loss values, indicating that synthetic embeddings increasingly approximate the real input data. This observation implies that cosine loss enhances the generator's proficiency in accurately reconstructing inputs, thereby improving the overall fidelity of the generated embeddings.

Figure 4 shows that the word embeddings derived from RoBERTa-large demonstrate a more consistent convergence in comparison to those of BERT-large during the training phase. This observation suggests that RoBERTa-large embeddings are more appropriate for training VQ-TEGAN, with the potential to produce embeddings that are semantically richer than those obtained from BERT-large embeddings.

### 4.3 Classification Performance in Few-shot Learning

Table 1 provides a comprehensive analysis of the efficacy of various data augmentation methods, namely EDA, EmbedHalluc, and VQ-TEGAN, when implemented in few-shot learning scenarios using RoBERTa-large and BERT-large models across nine distinct tasks. The hyperparameters for few-shot learning are presented in Appendix B, while the benchmark methods are described in Appendix C. The findings indicate that VQ-TEGAN consistently surpasses the other methods in most tasks, underscoring its robustness in text data augmentation. In particular, VQ-TEGAN significantly outperforms in seven tasks, with the exception of QNLI and RTE. However, VQ-TEGAN still achieves parity with EmbedHalluc on QNLI and is only 0.72% less accurate than EDA on RTE.

Although VQ-TEGAN demonstrates enhancements in RoBERTa-large, its performance remains comparable to other benchmarks when evaluated with BERT-large. EDA exhibits superior performance in MRPC (F1) with a score of 1.52, whereas EmbedHalluc surpasses in MNLI-mm, RTE, and WNLI by margins of 0.04%, 0.08%, and 0.94%, respectively. VQ-TEGAN also shows improved results, albeit marginally, with an increase of 0.08 in CoLA (Matt.), and 0.38%, 0.02%, and 1.94% in SST-2, MNLI-m, and QNLI, respectively. It is important to note that no significant performance disparities are observed when these models are applied to BERT-large.

In conclusion, VQ-TEGAN consistently surpasses EDA and EmbedHalluc, particularly when integrated with RoBERTa-large as opposed to BERT-large. The magnitude and complexity of the word embeddings of the employed PLM can significantly influence the extent of performance enhancement achieved with VQ-TEGAN. Given that VQ-TEGAN is trained directly on the word embeddings of the PLM, the utilization of more diverse and intricate embeddings for training culminates in more effective data augmentation. Consequently, VQ-TEGAN can be seen as a suitable data augmentation method to enhance the performance of larger PLMs relative to smaller ones.

### 4.4 Semantic Analysis on Mixup Embedding

Table 2 presents the three most prominent words decoded from the word embeddings of RoBERTa-large and BERT-large, demonstrating the highest cosine similarity to the mixup embeddings with different mixup ratio, $\lambda$. The words "beautiful", "bad", "characters", and "doubts" are used as input, and the results illustrate the alterations in embeddings under varying degrees of mixup. Note that the embeddings are congruent with the real embedding at $\lambda = 1.0$. The result is a representation of the words in the embeddings that demonstrate the highest cosine similarity to the real embedding for each PLM. The results show that the embeddings of all terms exhibit the highest cosine similarity to the embeddings of synonyms or capitalized forms for both PLMs.

In the case of the RoBERTa-large model, the list of closest word embeddings from $\lambda = 0.8$ is identical or slightly modified from $\lambda = 1.0$, including minor modifications to words (e.g., "suspicions") or capitalization (e.g., "BAD"). That is, the semantic properties of the closest embeddings exhibit minimal variation relative to the case with $\lambda = 1.0$. When $\lambda$ is set to 0.6, new words different from the list of $\lambda = 1.0$ start to appear in the third rank (e.g., "magnificent" for the word "beautiful" and "lousy" for the word "bad"). As $\lambda$ decreases to 0.4, many words that have similar semantic properties emerge in the list (e.g., "crappy" for the word "bad" and "protagonists" and "superheroes" for the word "characters"). This phenomenon is strengthened when $\lambda$ decreases to 0.2, showing an increasing deviation from the original words. For instance, the top three words decoded from the RoBERTa-large are "superheroes", "mystic", and "villan" for the word "characters". For a $\lambda$ of 0.0, the embedding is populated with novel words that are not related to the original words. It emphasizes the necessity of

| Model | CoLA (Matt.) | SST-2 (acc) | MRPC (F1) | QQP (acc) | MNLI-m (acc) | MNLI-mm (acc) | QNLI (acc) | RTE (acc) | WNLI (acc) |
|---|---|---|---|---|---|---|---|---|---|
| **RoBERTa-large** | 17.20±10.28 | 72.58±9.59 | 67.86±7.83 | 62.26±6.91 | 33.62±0.70 | 34.78±0.58 | 47.80±1.49 | 49.68±1.22 | 57.38±5.20 |
| +EDA | 12.42±6.78 | 70.48±6.78 | 68.68±13.98 | 57.22±18.77 | 33.78±1.22 | 33.88±2.34 | 49.22±1.24 | **50.96**±0.72 | 53.56±5.96 |
| +EmbedHalluc | 21.90±8.57 | 75.82±6.48 | 69.52±4.77 | 63.12±4.89 | 33.38±1.14 | 34.96±0.85 | **49.64**±0.75 | 49.54±1.01 | 55.32±8.20 |
| +**VQ-TEGAN** | **29.66**±8.02 | **78.14**±8.13 | **72.50**±4.16 | **70.98**±8.10 | **34.68**±1.14 | **36.00**±2.51 | **49.64**±1.40 | 50.24±0.42 | **62.60**±2.95 |
| **BERT-large** | 8.18±4.04 | 75.36±8.16 | 64.42±14.51 | 59.12±8.69 | 32.32±1.00 | 33.46±2.29 | 48.92±1.66 | 49.56±0.43 | 47.80±9.28 |
| +EDA | 10.48±3.59 | 78.22±4.36 | **73.14**±6.50 | 46.12±13.08 | 32.56±1.06 | 32.42±1.59 | 49.84±2.95 | 49.62±1.58 | 52.46±9.70 |
| +EmbedHalluc | 12.30±7.19 | 74.10±7.56 | 63.84±16.07 | 59.26±4.70 | 34.30±1.75 | **35.12**±2.21 | 48.60±2.30 | **49.64**±0.73 | **53.68**±8.11 |
| +**VQ-TEGAN** | **12.38**±4.53 | **78.60**±4.38 | 71.62±6.92 | **66.98**±5.59 | **34.32**±1.18 | 35.08±2.78 | **51.78**±1.27 | 49.56±0.50 | 52.74±6.77 |

Table 1: A comparative analysis of Conventional Fine-tuning, EDA, EmbedHalluc, and VQ-TEGAN, using RoBERTa-large and BERT-large as base models. The superior performance for each task is denoted in bold.

| Word Embedding | | RoBERTa-large | | | | BERT-large | | | |
|---|---|---|---|---|---|---|---|---|---|
| $\lambda$ | Rank | beautiful | bad | characters | doubts | beautiful | bad | characters | doubts |
| 1.0 | 1 | beautiful | bad | characters | doubts | beautiful | bad | characters | doubts |
| | 2 | gorgeous | Bad | character | doubt | gorgeous | good | character | doubted |
| | 3 | lovely | terrible | Characters | doubted | lovely | badly | protagonists | doubt |
| 0.8 | 1 | beautiful | bad | characters | doubts | beautiful | bad | characters | doubts |
| | 2 | gorgeous | Bad | character | doubted | gorgeous | badly | character | doubted |
| | 3 | lovely | BAD | Characters | suspicions | lovely | 295 | protagonists | doubt |
| 0.6 | 1 | beautiful | bad | characters | doubts | beautiful | bad | characters | doubts |
| | 2 | gorgeous | BAD | character | doubted | gorgeous | 295 | protagonists | [unused306] |
| | 3 | magnificent | lousy | Characters | suspicions | 1738 | 321 | 1743 | [unused298] |
| 0.4 | 1 | beautiful | bad | characters | doubts | 1736 | 1736 | 1736 | doubts |
| | 2 | gorgeous | lousy | protagonists | doubted | 1732 | 276 | 1743 | [unused659] |
| | 3 | magnificent | crappy | superheroes | suspicions | 1738 | 326 | 1732 | [unused276] |
| 0.2 | 1 | Beautiful | intertwined | superheroes | doubts | 1736 | 1736 | 1736 | [unused659] |
| | 2 | magnificent | sandy | mystic | timid | 1732 | 276 | 1743 | [unused80] |
| | 3 | the | crafted | vilains | dismay | 1743 | 1732 | 1732 | [unused176] |
| 0.0 | 1 | ACE | unfold | mystic | mystic | 1736 | 1736 | 1736 | [unused659] |
| | 2 | Apex | crafted | wretched | wretched | 1732 | 1732 | 1732 | [unused80] |
| | 3 | EA | intertwined | timid | timid | 45th | 45th | 45th | [unused176] |

Table 2: The top three words decoded from word embeddings in RoBERTa-large and BERT-large, exhibiting the highest degree of cosine similarity to the mixup embeddings with different $\lambda$.

the mixup for the augmentation via VQ-TEGAN.

In the case of the BERT-large model, at $\lambda = 0.8$, a minor change is observed for the word "bad", but no change is observed for other words. Interestingly, the new words included in the word "bad" include the semantically unrelated word "295". As $\lambda$ decreases to 0.6, there is a significant increase in unrelated tokens and numbers observed, indicating a stronger deviation from the original words. As the value of $\lambda$ is reduced from 0.4 to 0.0, the list is filled with semantically irrelevant words.

Our analysis indicates that as $\lambda$ decreases, the mixup embeddings exhibit an increasing divergence from the original words. Furthermore, the mixup embeddings produced by RoBERTa-large are observed to encapsulate more semantically rich and contextually pertinent words at smaller $\lambda$ compared to those generated by BERT-large. This observation suggests that the mixup embeddings of RoBERTa-large maintain a higher degree of semantic coherence under mixup conditions compared to BERT-large. This is corroborated by the classification performance presented in Table 1, which demonstrates that RoBERTa-large exhibits a significant improvement in performance with mixup embeddings, whereas BERT-large does not show a comparable enhancement.

In conclusion, when VQ-TEGAN generates meaningful synthetic embeddings and integrates mixup embeddings with real embeddings for few-shot learning, it has the potential to facilitate the application of mixup embeddings with an expanded and more heterogeneous semantic spectrum for few-shot learning. Additional semantic analysis on mixup embeddings can be found in the Appendix D.

### 4.5 Sensitivity Analysis on Mixup Ratio

In Table 3, we present a comparative analysis of the results derived from conventional fine-tuning and our proposed model, employing three distinct $\lambda$ values (0.0, 0.2, and 0.4). The scenario with $\lambda = 1$ was omitted from the sensitivity analysis due to its redundancy in merely duplicating the real embedding. Likewise, scenarios with $\lambda = 0.6$ and $\lambda = 0.8$ were excluded as their results did not show significant deviations from those presented in Table 2.

| Model | CoLA (Matt.) | SST-2 (acc) | MRPC (F1) | QQP (acc) | MNLI-m (acc) | MNLI-mm (acc) | QNLI (acc) | RTE (acc) | WNLI (acc) |
|---|---|---|---|---|---|---|---|---|---|
| **RoBERTa-large** | 17.20±10.28 | 72.58±9.59 | 67.86±7.83 | 62.26±6.91 | 33.62±0.70 | 34.78±0.58 | 47.80±1.49 | 49.68±1.22 | 57.38±5.20 |
| w/ $\lambda = 0.0$ | **28.32**±11.60 | **78.14**±8.13 | **71.98**±6.36 | **70.98**±8.10 | **34.60**±1.56 | **36.00**±2.51 | **48.22**±1.42 | **50.24**±0.42 | **60.96**±4.42 |
| w/ $\lambda = 0.2$ | **29.66**±8.02 | **76.84**±5.88 | **72.50**±4.16 | **66.68**±7.68 | **34.40**±0.90 | 34.66±1.07 | **49.64**±1.40 | **50.02**±0.65 | **59.18**±4.08 |
| w/ $\lambda = 0.4$ | **18.30**±3.72 | **74.42**±7.33 | **71.62**±6.86 | **65.24**±9.24 | **34.68**±1.14 | **35.60**±2.29 | **48.94**±0.27 | **50.22**±0.76 | **62.60**±2.95 |
| **BERT-large** | 8.18±4.04 | 75.36±8.16 | 64.42±14.51 | 59.12±8.69 | 32.32±1.00 | 33.46±2.29 | 48.92±1.66 | 49.56±0.43 | 47.80±9.28 |
| w/ $\lambda = 0.0$ | **9.44**±6.84 | **77.34**±5.00 | **68.20**±11.32 | **66.98**±5.59 | **34.32**±1.18 | **35.08**±2.78 | **50.26**±1.81 | 49.42±0.44 | **52.74**±6.77 |
| w/ $\lambda = 0.2$ | **12.38**±4.53 | **77.00**±5.36 | **71.62**±6.92 | **62.76**±12.57 | **33.46**±1.57 | **34.24**±1.41 | **50.12**±0.98 | 49.46±0.91 | **51.22**±7.04 |
| w/ $\lambda = 0.4$ | **9.62**±7.42 | **78.60**±4.38 | **69.96**±7.08 | **63.78**±7.14 | **33.58**±1.65 | **34.02**±3.10 | **51.78**±1.27 | **49.56**±0.50 | **52.34**±6.52 |

Table 3: A comparative analysis of conventional fine-tuning and VQ-TEGAN for different $\lambda$, using RoBERTa-large and BERT-large as base models. The bold numbers indicate instances where VQ-TEGAN outperforms conventional fine-tuning for each respective task, while underlined numbers indicate the highest performance.

Using RoBERTa-large for few-shot learning, VQ-TEGAN demonstrates superior performance relative to fine-tuning across all evaluated tasks. In general, $\lambda = 0.0$ and $\lambda = 0.2$ exhibit increased efficacy compared to traditional fine-tuning and $\lambda = 0.4$, with the exception of MNLI-m and WNLI. Specifically, for tasks such as SST-2, QQP, MNLI-mm, and RTE, the optimal results are observed with $\lambda = 0.0$. In contrast, $\lambda = 0.2$ achieves superior results in CoLA, MRPC, and QNLI. In particular, $\lambda = 0.4$ surpasses $\lambda = 0.0$ and $\lambda = 0.2$ exclusively in MNLI-m and WNLI. These findings indicate that the incorporation of synthetic embeddings or mixup embeddings significantly enhances model generalization and performance.

In contrast, using BERT-large for few-shot learning reveals a distinct pattern. Specifically, a $\lambda$ value of 0.2 enhances performance beyond traditional fine-tuning in the CoLA and MRPC datasets. The most substantial performance improvements are achieved with $\lambda = 0.4$ in the SST-2, QNLI, and RTE tasks. In particular, a $\lambda$ value of 0.0 yields the highest performance metrics in QQP, MNLI-m, MNLI-mm, and WNLI. These observations suggest that the efficacy of BERT is differentially influenced by varying $\lambda$ values and synthetic embeddings contingent on the specific task, thereby indicating the absence of a universally optimal $\lambda$ value across all tasks.

## 5 Conclusion

This study introduces VQ-TEGAN, a novel data augmentation method for text embedding. VQ-TEGAN generates embeddings across various semantic and synonymic dimensions of PLM embeddings, facilitating more efficient and effective acquisition of a broader spectrum of semantics during the fine-tuning of PLMs with limited training datasets. Our empirical analysis reveals that VQ-TEGAN (1) achieves superior performance enhancements on GLUE benchmark tasks in few-shot learning contexts, (2) is more compact and lightweight compared to other language models employed for data augmentation, (3) augments PLM performance, particularly when utilized with PLMs possessing larger embeddings, and (4) introduces a more efficient loss function for text embedding generation via the convergence of loss functions.

## 6 Limitations

Despite its novelty, there are limitations that need to be addressed in future work. As discussed in section 4.4, the semantic analysis of the closest PLM word embeddings to the mixup embeddings elucidates the potential for formulating a novel embedding space conducive to few-shot learning. However, a limitation is identified where VQ-TEGAN-generated embeddings may converge within a space similar to other semantic embeddings, attributable to the anisotropy issue inherent in PLM word embeddings (Ethayarajh, 2019; Li et al., 2020). A possible approach is to train VQ-TEGAN utilizing word embeddings derived from PLMs that have been refined through contrastive learning(Gao et al., 2021), addressing the anisotropy issue within the embedding space. Lastly, this study exclusively investigates the instances of VQ-TEGAN utilizing RoBERTa-large and BERT-large. For subsequent study, a broader spectrum of PLMs should be explored for the implementation of VQ-TEGAN.

## 7 Ethics Statement

This paper investigates data augmentation in the generation of embeddings for few-shot learning. It is not anticipated that this research will raise any ethical or social issues. All data utilized in this study is publicly accessible and has been utilized by numerous researchers. The proposed method does not introduce any ethical biases into the data.

## References


Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7383–7390.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR.

Farid Arthaud, Rachel Bawden, and Alexandra Birch. 2021. Few-shot learning through contextual data augmentation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1049–1062, Online. Association for Computational Linguistics.

Bjorn Barz and Joachim Denzler. 2020. Deep learning on small datasets without pre-training using cosine loss. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1371–1380.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Thomas Body, Xiaohui Tao, Yuefeng Li, Lin Li, and Ning Zhong. 2021. Using back-and-forth translation to create artificial augmented textual data for sentiment analysis models. *Expert Systems with Applications*, 178:115033.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Rakesh Chada and Pradeep Natarajan. 2021. Fewshotqa: A simple framework for few-shot learning of question answering tasks using pre-trained text-to-text models. *arXiv preprint arXiv:2109.01951*.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020a. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. *arXiv preprint arXiv:2004.12239*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020b. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Thomas Dopierre, Christophe Gravier, and Wilfried Logerais. 2021. Protaugment: Intent detection meta-learning through unsupervised diverse paraphrasing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2454–2466. Association for Computational Linguistics.

Zach Eaton-Rosen, Felix J. S. Bragman, Sébastien Ourselin, and M. Jorge Cardoso. 2018. Improving data augmentation for medical image segmentation. In *OpenReview*.

Yousef El-Laham and Svitlana Vyetrenko. 2022. Styletime: Style transfer for synthetic time series generation. In *Proceedings of the Third ACM International Conference on AI in Finance*, pages 489–496.

Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *arXiv preprint arXiv:1909.00512*.

Elizabeth Fons, Paula Dawson, Xiao-jun Zeng, John Keane, and Alexandros Iosifidis. 2020. Evaluating data augmentation for financial time series classification. *arXiv preprint arXiv:2010.15111*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*.

Aakriti Gupta, Kapil Thadani, and Neil O'Hare. 2020. Effective few-shot classification with transfer learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1061–1066.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. arxiv 2015. *arXiv preprint arXiv:1508.01991*.

Ouyang Jia, Huimin Huang, Jiaxin Ren, Luodi Xie, and Yinyin Xiao. 2023. Contrastive learning with text augmentation for text classification. *Applied Intelligence*, pages 1–10.

Yiren Jian, Chongyang Gao, and Soroush Vosoughi. 2022. Embedding hallucination for few-shot language fine-tuning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3, pages 5522–5530. Association for Computational Linguistics.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. AEDA: An easier data augmentation technique for text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2748–2754, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Justin Ker, Lipo Wang, Jai Rao, and Tchoyoson Lim. 2017. Deep learning applications in medical image analysis. *Ieee Access*, 6:9375–9389.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information*, 10(4):150.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*.

Yang Liu, Zhenyue Qin, Saeed Anwar, Pan Ji, Dongwoo Kim, Sabrina Caldwell, and Tom Gedeon. 2021. Invertible denoising network: A light solution for real noise removal. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13365–13374.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Shayne Longpre, Yu Wang, and Christopher DuBois. 2020. How effective is task-agnostic data augmentation for pretrained transformers? *arXiv preprint arXiv:2010.01764*.

Nikolaos Malandrakis, Minmin Shen, Anuj Goyal, Shuyang Gao, Abhishek Sethi, and Angeliki Metallinou. 2019. Controlled text generation for data augmentation in intelligent artificial agents. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 90–98, Hong Kong. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. 2020. Ssmba: Self-supervised manifold based data augmentation for improving out-of-domain robustness. *arXiv preprint arXiv:2009.10195*.

Aytuğ Onan. 2023. Gtr-ga: Harnessing the power of graph-based neural networks and genetic algorithms for text augmentation. *Expert Systems with Applications*, page 120908.

Lucas Francisco Amaral Orosco Pellicer, Taynan Maier Ferreira, and Anna Helena Reali Costa. 2023. Data augmentation techniques in natural language processing. *Applied Soft Computing*, 132:109803.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Frédéric Piedboeuf and Philippe Langlais. 2022. Effective data augmentation for sentence classification using one vae per class. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3454–3464.

Lutz Prechelt. 2002. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Faisal Rahutomo, Teruaki Kitasuka, and Masayoshi Aritsugi. 2012. Semantic cosine similarity. In *The 7th international student conference on advanced science and technology ICAST*, volume 4, page 1.

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. 2021. Few-shot question answering by pretraining span selection. *arXiv preprint arXiv:2101.00438*.

Marc'Aurelio Ranzato, Fu Jie Huang, Y-Lan Boureau, and Yann LeCun. 2007. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

William Yang Wang and Diyi Yang. 2015. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2557–2563.

Jason Wei, Chengyu Huang, Soroush Vosoughi, Yu Cheng, and Shiqi Xu. 2021. Few-shot text classification with triplet networks, data augmentation, and curriculum learning. *arXiv preprint arXiv:2103.07552*.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Haiyan Wu, Yanyun Qu, Shaohui Lin, Jian Zhou, Ruizhi Qiao, Zhizhong Zhang, Yuan Xie, and Lizhuang Ma. 2021. Contrastive learning for compact single image dehazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10551–10560.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268.

Ying Yu, Bingying Tang, Ronglai Lin, Shufa Han, Tang Tang, and Ming Chen. 2019. Cwgan: Conditional wasserstein generative adversarial nets for fault data generation. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2713–2718.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Le Zhang, Zichao Yang, and Diyi Yang. 2022. Treemix: Compositional constituency-based data augmentation for natural language understanding. *arXiv preprint arXiv:2205.06153*.

Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. 2016. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1):47–57.

Wanjun Zhong, Duyu Tang, Jiahai Wang, Jian Yin, and Nan Duan. 2021. Useradapter: Few-shot user learning in sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1484–1488.

# A Training Details for VQ-TEGAN

The generator architecture includes an encoder, a decoder, and a codebook of latent vectors. The encoder is composed of four sequential blocks, each containing a fully connected layer, batch normalization, and the LeakyReLU activation function (Jian et al., 2022). This encoder progressively reduces the dimensionality to 1024, 512, 256, and 128. The codebook comprises quantized latent vectors that correspond to the output dimensions of the encoder. The quantity of codebook vectors is adjusted as a hyperparameter during the experimental procedures. The decoder, which structurally parallels the encoder, consists of four blocks that expand the quantized codebook vectors to dimensions of 128, 256, 512, and 1024. The discriminator is structured with three blocks, having dimensions of 512, 512, and 1, respectively, and produces a singular tensor output. VQ-TEGAN is subjected to training for 10 epochs with a batch size of 64, utilizing the Adam optimizer ($\beta = (0.5, 0.999)$) and a fixed random seed of 42. The training process includes a grid search for the learning rates of $2e^{-5}$ and $5e^{-5}$, as well as codebook vector quantities of 1024, 2048, and 4096.

# B Hyperparameters for Few-shot Learning

The model is trained using learning rates of $1e^{-5}$ and $2e^{-5}$, with batch sizes of 4 and 8. Random number generation seeds of 13, 21, 42, 87, and 100 are utilized. The training process was capped at 150 epochs, with the final model being selected based on validation accuracy at each epoch. An early stopping mechanism is used to mitigate overfitting, halting training if no improvement in validation accuracy is observed after 100 epochs (Prechelt, 2002).

To train the PLM with augmented embeddings, comprehensive experiments are conducted across all parameters. The mixup ratios for $x$ and $\hat{x}$ are evaluated at $\lambda$ values of 0, 0.2, and 0.4 as illustrated in Eq. 6. Both EDA and EmbedHalluc are executed using default settings, with EDA's data augmentation further explored by generating 4 and 9 additional samples.

The algorithms are implemented using Python 3.10.8 and PyTorch 1.13.1. The experiments are carried out on an Ubuntu 20.04.6 system equipped with a Nvidia RTX 3090 TI (24 GB RAM) and an Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz.

The NLTK 3.8.1 toolkit is used for synonym replacement in the EDA process. RoBERTa-large and BERT-large models, along with their tokenizers, are sourced from the Hugging Face Transformers library.

## C Benchmarks

The performance of VQ-TEGAN is evaluated in comparison to established benchmarks: conventional fine-tuning, EDA, and EmbedHalluc based on cWGAN.

- **Conventional Fine-tuning** constitutes a fundamental approach where a few-shot language model is trained exclusively on the provided dataset, devoid of any supplementary data augmentation.
- **EDA**(Wei and Zou, 2019) represents a data augmentation that incorporates four principal techniques: synonym replacement, random deletion, random swap, and random addition. This method is both intuitive and efficient, facilitating the generation of a substantial number of synthetic sentences in a straightforward manner.
- **EmbedHalluc**(Jian et al., 2022) leverages cWGAN to augment textual data within the embedding space. The training process encompasses both the generator and the discriminator, with data augmentation being realized by doubling the few-shot data via the generator.

## D Text Embedding Analysis

Tables 4, 5, 6, and 7 present the words of the embedding tokens that exhibit the three highest cosine similarities between the mixup embeddings generated by VQ-TEGAN and the embeddings within the PLM under various parameter configurations. Tables 4 and 5 elucidate the results for RoBERTa-large embeddings, while Tables 6 and 7 illustrate the results for BERT-large embeddings. The parameter configurations encompass the learning rate of VQ-TEGAN, the number of codebook vectors, and five distinct values of $\lambda$.

Tables 4 and 5 show the evolving patterns in the semantic representations of the embeddings as the parameter $\lambda$ chages, as discerned through our comprehensive analysis.

At a $\lambda$ value of 0.0, where the embeddings are synthesized exclusively by VQ-TEGAN in the absence of any real embeddings, the cosine similarity fails to effectively discern relatedness or synonymy. This observation implies that the synthetic embeddings may exhibit abstract or non-traditional associations at this $\lambda$, which deviate from the conventional semantic relationships observed in real embeddings.

A notable change is observed when the parameter $\lambda$ is elevated to 0.2 or 0.4. At these $\lambda$ values, the top three synonyms for each text sample exhibit increased diversity, which means that the mixed embeddings now encapsulate a wider spectrum of semantic similarities. For instance, Table 4 demonstrates that the mixup embedding of “beautiful”, with a $\lambda$ of 0.2, achieves the highest cosine similarities of 0.751, 0.742, and 0.740 with the embeddings of adjectives bearing analogous meanings, such as “exquisite”, “magnificent”, and “marvellous”, respectively, for $\mathrm{lr}_{VQ}$ of 5e-05 and a codebook size of 4096. In Table 5, it is evident that the mixup embedding of “characters” manifests the three highest cosine similarities of 0.670, 0.670, and 0.658 with the embeddings of nouns possessing similar or identical meanings, such as “villains”, “superheroes”, and “characters”. This observation is pivotal, as it suggests that the mixup embedding at these $\lambda$ values transcends a mere replication of the original meanings. Moreover, it introduces an extensive array of related concepts with real embeddings. The inclusion of 20% of $x$ functions as an anchor, anchoring the synthetic embedding within the original semantic framework while still allowing the introduction of novel nuances. This equilibrium, facilitated by synthetic embeddings generated by VQ-TEGAN, which adeptly constructs the semantic space of PLM's embedding, culminates in mixup embeddings that are enriched with supplementary contextual meaning. This generates a more nuanced and comprehensive semantic comprehension.

Upon observation, it was observed that when the parameter $\lambda$ exceeds the threshold of 0.6, the augmented embeddings exhibit a pronounced resemblance to the real embedding $x$. This phenomenon indicates that at elevated values, the mixup embeddings converge more closely with the semantic attributes of the genuine embedding, thereby diminishing the distinctions from the original embedding. As a result, this convergence may precipitate issues of data redundancy, as the mixup embeddings may not provide substantially novel or diverse information relative to the original dataset.

Tables 6 and 7 demonstrate that the evaluation of BERT-large mixup embeddings via cosine sim-

Table 4: The text and cosine similarity metrics of the three words decoded from RoBERTa-large, exhibiting the highest cosine similarity between the word embeddings in RoBERTa-large and the mixup embeddings using the adjectives 'beautiful' and 'bad'. The parameter $\mathrm{lr}_{VQ}$ denotes the learning rate of the VQ-TEGAN, while the term codebook refers to the number of codebook vectors in VQ-TEGAN ($K$ of $\mathcal{Z} = \{z_k\}_{k=1}^{K}$ as specified in Eq. 1). Additionally, $\lambda$ represents the rate of authentic text embedding.

| Parameter | | Text | **beautiful** | | | **bad** | | |
|---|---|---|---|---|---|---|---|---|
| $\mathrm{lr}_{VQ}$ | Codebook | $\lambda$ | Rank 1 | Rank 2 | Rank 3 | Rank 1 | Rank 2 | Rank 3 |
| 2e-05 | 1024 | 0.0 | ACE$_{(0.691)}$ | Apex$_{(0.688)}$ | EA$_{(0.687)}$ | unfold$_{(0.606)}$ | crafted$_{(0.602)}$ | intertwined$_{(0.601)}$ |
| | | 0.2 | Beautiful$_{(0.702)}$ | magnificent$_{(0.701)}$ | the$_{(0.697)}$ | intertwined$_{(0.621)}$ | sandy$_{(0.619)}$ | crafted$_{(0.618)}$ |
| | | 0.4 | beautiful$_{(0.83)}$ | gorgeous$_{(0.769)}$ | magnificent$_{(0.764)}$ | bad$_{(0.732)}$ | lousy$_{(0.660)}$ | crappy$_{(0.658)}$ |
| | | 0.6 | beautiful$_{(0.935)}$ | gorgeous$_{(0.834)}$ | lovely$_{(0.790)}$ | bad$_{(0.882)}$ | BAD$_{(0.683)}$ | lousy$_{(0.682)}$ |
| | | 0.8 | beautiful$_{(0.987)}$ | gorgeous$_{(0.854)}$ | lovely$_{(0.804)}$ | bad$_{(0.974)}$ | Bad$_{(0.713)}$ | BAD$_{(0.707)}$ |
| | 2048 | 0.0 | extravagant$_{(0.721)}$ | stereotypical$_{(0.718)}$ | astounding$_{(0.718)}$ | extravagant$_{(0.721)}$ | stereotypical$_{(0.718)}$ | astounding$_{(0.718)}$ |
| | | 0.2 | exquisite$_{(0.751)}$ | magnificent$_{(0.742)}$ | astounding$_{(0.740)}$ | obnoxious$_{(0.728)}$ | extravagant$_{(0.723)}$ | stereotypical$_{(0.723)}$ |
| | | 0.4 | beautiful$_{(0.819)}$ | magnificent$_{(0.787)}$ | gorgeous$_{(0.781)}$ | bad$_{(0.815)}$ | crappy$_{(0.731)}$ | horrendous$_{(0.731)}$ |
| | | 0.6 | beautiful$_{(0.923)}$ | gorgeous$_{(0.840)}$ | magnificent$_{(0.799)}$ | bad$_{(0.922)}$ | BAD$_{(0.728)}$ | terrible$_{(0.719)}$ |
| | | 0.8 | beautiful$_{(0.983)}$ | gorgeous$_{(0.860)}$ | lovely$_{(0.806)}$ | bad$_{(0.983)}$ | BAD$_{(0.721)}$ | Bad$_{(0.719)}$ |
| | 4096 | 0.0 | Shoes$_{(0.702)}$ | Changes$_{(0.702)}$ | Moments$_{(0.695)}$ | hateful$_{(0.725)}$ | despicable$_{(0.722)}$ | wretched$_{(0.718)}$ |
| | | 0.2 | Shoes$_{(0.703)}$ | Hair$_{(0.700)}$ | Awesome$_{(0.699)}$ | horrendous$_{(0.749)}$ | hateful$_{(0.741)}$ | despicable$_{(0.740)}$ |
| | | 0.4 | Beautiful$_{(0.758)}$ | beautiful$_{(0.743)}$ | gorgeous$_{(0.707)}$ | bad$_{(0.866)}$ | BAD$_{(0.749)}$ | horrendous$_{(0.746)}$ |
| | | 0.6 | beautiful$_{(0.891)}$ | gorgeous$_{(0.807)}$ | Beautiful$_{(0.779)}$ | bad$_{(0.950)}$ | BAD$_{(0.749)}$ | terrible$_{(0.734)}$ |
| | | 0.8 | beautiful$_{(0.976)}$ | gorgeous$_{(0.852)}$ | lovely$_{(0.797)}$ | bad$_{(0.99)}$ | BAD$_{(0.725)}$ | terrible$_{(0.719)}$ |
| 5e-05 | 1024 | 0.0 | demonic$_{(0.637)}$ | anarchist$_{(0.630)}$ | superhero$_{(0.626)}$ | demonic$_{(0.637)}$ | anarchist$_{(0.630)}$ | superhero$_{(0.626)}$ |
| | | 0.2 | marvelous$_{(0.683)}$ | majestic$_{(0.680)}$ | magnificent$_{(0.680)}$ | demonic$_{(0.674)}$ | hateful$_{(0.668)}$ | heinous$_{(0.668)}$ |
| | | 0.4 | beautiful$_{(0.822)}$ | gorgeous$_{(0.763)}$ | magnificent$_{(0.748)}$ | bad$_{(0.811)}$ | BAD$_{(0.700)}$ | horrendous$_{(0.691)}$ |
| | | 0.6 | beautiful$_{(0.932)}$ | gorgeous$_{(0.832)}$ | lovely$_{(0.788)}$ | bad$_{(0.929)}$ | BAD$_{(0.730)}$ | terrible$_{(0.715)}$ |
| | | 0.8 | beautiful$_{(0.986)}$ | gorgeous$_{(0.854)}$ | lovely$_{(0.804)}$ | bad$_{(0.986)}$ | BAD$_{(0.720)}$ | terrible$_{(0.714)}$ |
| | 2048 | 0.0 | exquisite$_{(0.647)}$ | extravagant$_{(0.636)}$ | ludicrous$_{(0.633)}$ | exquisite$_{(0.647)}$ | extravagant$_{(0.636)}$ | ludicrous$_{(0.633)}$ |
| | | 0.2 | exquisite$_{(0.718)}$ | extravagant$_{(0.680)}$ | astounding$_{(0.679)}$ | troublesome$_{(0.675)}$ | ludicrous$_{(0.671)}$ | exquisite$_{(0.670)}$ |
| | | 0.4 | beautiful$_{(0.789)}$ | exquisite$_{(0.762)}$ | magnificent$_{(0.750)}$ | bad$_{(0.771)}$ | horrendous$_{(0.706)}$ | dreadful$_{(0.703)}$ |
| | | 0.6 | beautiful$_{(0.915)}$ | gorgeous$_{(0.817)}$ | magnificent$_{(0.781)}$ | bad$_{(0.909)}$ | terrible$_{(0.712)}$ | horrendous$_{(0.703)}$ |
| | | 0.8 | beautiful$_{(0.982)}$ | gorgeous$_{(0.851)}$ | lovely$_{(0.798)}$ | bad$_{(0.981)}$ | terrible$_{(0.717)}$ | horrible$_{(0.703)}$ |
| | 4096 | 0.0 | esoteric$_{(0.705)}$ | clandestine$_{(0.698)}$ | subversive$_{(0.697)}$ | esoteric$_{(0.705)}$ | clandestine$_{(0.698)}$ | subversive$_{(0.697)}$ |
| | | 0.2 | exquisite$_{(0.739)}$ | magnificent$_{(0.724)}$ | marvelous$_{(0.723)}$ | nefarious$_{(0.720)}$ | obnoxious$_{(0.713)}$ | crappy$_{(0.712)}$ |
| | | 0.4 | beautiful$_{(0.800)}$ | magnificent$_{(0.774)}$ | exquisite$_{(0.769)}$ | bad$_{(0.788)}$ | crappy$_{(0.736)}$ | lousy$_{(0.734)}$ |
| | | 0.6 | beautiful$_{(0.911)}$ | gorgeous$_{(0.829)}$ | magnificent$_{(0.794)}$ | bad$_{(0.907)}$ | BAD$_{(0.746)}$ | crappy$_{(0.726)}$ |
| | | 0.8 | beautiful$_{(0.979)}$ | gorgeous$_{(0.857)}$ | lovely$_{(0.807)}$ | bad$_{(0.979)}$ | BAD$_{(0.733)}$ | Bad$_{(0.724)}$ |
| | | 1.0 | beautiful$_{(1.000)}$ | gorgeous$_{(0.846)}$ | lovely$_{(0.792)}$ | bad$_{(1.000)}$ | Bad$_{(0.706)}$ | terrible$_{(0.691)}$ |

Table 5: The text and cosine similarity metrics of the three words decoded from RoBERTa-large, exhibiting the highest cosine similarity between the word embeddings in RoBERTa-large and the mixup embeddings using the nouns 'characters' and 'doubts'

| Parameter | | Text | **characters** | | | **doubts** | | |
|---|---|---|---|---|---|---|---|---|
| $lr_{VQ}$ | Codebook | $\lambda$ | Rank 1 | Rank 2 | Rank 3 | Rank 1 | Rank 2 | Rank 3 |
| 2e-05 | 1024 | 0.0 | mystic$_{(0.719)}$ | wretched$_{(0.718)}$ | timid$_{(0.715)}$ | mystic$_{(0.719)}$ | wretched$_{(0.718)}$ | timid$_{(0.715)}$ |
| | | 0.2 | superheroes$_{(0.719)}$ | mystic$_{(0.716)}$ | villains$_{(0.716)}$ | doubts$_{(0.752)}$ | timid$_{(0.725)}$ | dismay$_{(0.722)}$ |
| | | 0.4 | characters$_{(0.826)}$ | protagonists$_{(0.743)}$ | superheroes$_{(0.740)}$ | doubts$_{(0.877)}$ | doubted$_{(0.742)}$ | suspicions$_{(0.736)}$ |
| | | 0.6 | characters$_{(0.934)}$ | character$_{(0.744)}$ | Characters$_{(0.744)}$ | doubts$_{(0.954)}$ | doubted$_{(0.751)}$ | suspicions$_{(0.745)}$ |
| | | 0.8 | characters$_{(0.987)}$ | character$_{(0.764)}$ | Characters$_{(0.738)}$ | doubts$_{(0.991)}$ | doubted$_{(0.736)}$ | suspicions$_{(0.730)}$ |
| | 2048 | 0.0 | extravagant$_{(0.721)}$ | stereotypical$_{(0.718)}$ | astounding$_{(0.718)}$ | extravagant$_{(0.721)}$ | stereotypical$_{(0.718)}$ | astounding$_{(0.718)}$ |
| | | 0.2 | stereotypical$_{(0.724)}$ | extravagant$_{(0.719)}$ | imaginative$_{(0.713)}$ | doubtful$_{(0.722)}$ | doubts$_{(0.718)}$ | bogus$_{(0.718)}$ |
| | | 0.4 | characters$_{(0.787)}$ | protagonists$_{(0.729)}$ | protagonist$_{(0.715)}$ | doubts$_{(0.852)}$ | doubted$_{(0.752)}$ | doubtful$_{(0.743)}$ |
| | | 0.6 | characters$_{(0.915)}$ | character$_{(0.743)}$ | protagonists$_{(0.738)}$ | doubts$_{(0.942)}$ | doubted$_{(0.762)}$ | suspicions$_{(0.737)}$ |
| | | 0.8 | characters$_{(0.982)}$ | character$_{(0.767)}$ | Characters$_{(0.730)}$ | doubts$_{(0.988)}$ | doubted$_{(0.744)}$ | suspicions$_{(0.729)}$ |
| | 4096 | 0.0 | hateful$_{(0.725)}$ | despicable$_{(0.722)}$ | wretched$_{(0.718)}$ | hateful$_{(0.725)}$ | despicable$_{(0.722)}$ | wretched$_{(0.718)}$ |
| | | 0.2 | demonic$_{(0.718)}$ | nonsensical$_{(0.716)}$ | despicable$_{(0.715)}$ | doubts$_{(0.752)}$ | doubtful$_{(0.739)}$ | bogus$_{(0.724)}$ |
| | | 0.4 | characters$_{(0.839)}$ | protagonists$_{(0.733)}$ | protagonist$_{(0.724)}$ | doubts$_{(0.887)}$ | doubted$_{(0.761)}$ | doubtful$_{(0.748)}$ |
| | | 0.6 | characters$_{(0.944)}$ | character$_{(0.755)}$ | protagonists$_{(0.729)}$ | doubts$_{(0.961)}$ | doubted$_{(0.760)}$ | suspicions$_{(0.738)}$ |
| | | 0.8 | characters$_{(0.989)}$ | character$_{(0.767)}$ | Characters$_{(0.727)}$ | doubts$_{(0.993)}$ | doubted$_{(0.738)}$ | suspicions$_{(0.725)}$ |
| 5e-05 | 1024 | 0.0 | boxes$_{(0.624)}$ | cats$_{(0.618)}$ | Mustang$_{(0.618)}$ | demonic$_{(0.637)}$ | anarchist$_{(0.630)}$ | superhero$_{(0.626)}$ |
| | | 0.2 | Characters$_{(0.703)}$ | characters$_{(0.699)}$ | protagonists$_{(0.676)}$ | doubts$_{(0.690)}$ | demonic$_{(0.663)}$ | baffled$_{(0.658)}$ |
| | | 0.4 | characters$_{(0.874)}$ | Characters$_{(0.762)}$ | protagonists$_{(0.720)}$ | doubts$_{(0.853)}$ | doubted$_{(0.712)}$ | suspicions$_{(0.703)}$ |
| | | 0.6 | characters$_{(0.960)}$ | Characters$_{(0.764)}$ | character$_{(0.747)}$ | doubts$_{(0.949)}$ | doubted$_{(0.738)}$ | suspicions$_{(0.729)}$ |
| | | 0.8 | characters$_{(0.993)}$ | character$_{(0.760)}$ | Characters$_{(0.741)}$ | doubts$_{(0.990)}$ | doubted$_{(0.730)}$ | suspicions$_{(0.724)}$ |
| | 2048 | 0.0 | exquisite$_{(0.647)}$ | extravagant$_{(0.636)}$ | ludicrous$_{(0.633)}$ | exquisite$_{(0.647)}$ | extravagant$_{(0.636)}$ | ludicrous$_{(0.633)}$ |
| | | 0.2 | exquisite$_{(0.664)}$ | extravagant$_{(0.656)}$ | outlandish$_{(0.654)}$ | ludicrous$_{(0.669)}$ | bogus$_{(0.667)}$ | exquisite$_{(0.666)}$ |
| | | 0.4 | characters$_{(0.766)}$ | protagonists$_{(0.676)}$ | protagonist$_{(0.663)}$ | doubts$_{(0.814)}$ | doubtful$_{(0.709)}$ | doubted$_{(0.699)}$ |
| | | 0.6 | characters$_{(0.911)}$ | Characters$_{(0.726)}$ | protagonists$_{(0.708)}$ | doubts$_{(0.932)}$ | doubted$_{(0.737)}$ | doubtful$_{(0.714)}$ |
| | | 0.8 | characters$_{(0.982)}$ | character$_{(0.760)}$ | Characters$_{(0.704)}$ | doubts$_{(0.987)}$ | doubted$_{(0.733)}$ | suspicions$_{(0.719)}$ |
| | 4096 | 0.0 | impoverished$_{(0.634)}$ | obsolete$_{(0.628)}$ | obscured$_{(0.627)}$ | impoverished$_{(0.634)}$ | obsolete$_{(0.628)}$ | obscured$_{(0.627)}$ |
| | | 0.2 | villains$_{(0.670)}$ | superheroes$_{(0.670)}$ | characters$_{(0.658)}$ | doubts$_{(0.727)}$ | disagreements$_{(0.674)}$ | shortcomings$_{(0.674)}$ |
| | | 0.4 | characters$_{(0.823)}$ | Characters$_{(0.726)}$ | protagonists$_{(0.707)}$ | doubts$_{(0.866)}$ | doubted$_{(0.716)}$ | suspicions$_{(0.712)}$ |
| | | 0.6 | characters$_{(0.933)}$ | Characters$_{(0.753)}$ | character$_{(0.726)}$ | doubts$_{(0.951)}$ | doubted$_{(0.737)}$ | suspicions$_{(0.733)}$ |
| | | 0.8 | characters$_{(0.987)}$ | character$_{(0.755)}$ | Characters$_{(0.742)}$ | doubts$_{(0.990)}$ | doubted$_{(0.730)}$ | suspicions$_{(0.725)}$ |
| | | 1.0 | characters$_{(1.000)}$ | character$_{(0.756)}$ | Characters$_{(0.711)}$ | doubts$_{(1.000)}$ | doubt$_{(0.718)}$ | doubted$_{(0.708)}$ |

Table 6: The text and cosine similarity metrics of the three words decoded from BERT-large, exhibiting the highest cosine similarity between the word embeddings in RoBERTa-large and the mixup embeddings using the adjectives ‘beautiful’ and ‘bad’.’

| Parameter | | Text | **beautiful** | | | **bad** | | |
|---|---|---|---|---|---|---|---|---|
| lr$_{VQ}$ | Codebook | $\lambda$ | Rank 1 | Rank 2 | Rank 3 | Rank 1 | Rank 2 | Rank 3 |
| 2e-05 | 1024 | 0.0 | 1736$_{(0.642)}$ | 1732$_{(0.630)}$ | 45th$_{(0.629)}$ | 1736$_{(0.642)}$ | 1732$_{(0.630)}$ | 45th$_{(0.629)}$ |
| | | 0.2 | 1736$_{(0.653)}$ | 1732$_{(0.641)}$ | 1743$_{(0.638)}$ | 1736$_{(0.651)}$ | 276$_{(0.640)}$ | 1732$_{(0.638)}$ |
| | | 0.4 | 1736$_{(0.650)}$ | 1732$_{(0.639)}$ | 1738$_{(0.637)}$ | 1736$_{(0.635)}$ | 276$_{(0.633)}$ | 326$_{(0.632)}$ |
| | | 0.6 | beautiful$_{(0.771)}$ | gorgeous$_{(0.616)}$ | 1738$_{(0.608)}$ | bad$_{(0.783)}$ | 295$_{(0.582)}$ | 321$_{(0.0.582)}$ |
| | | 0.8 | beautiful$_{(0.938)}$ | gorgeous$_{(0.656)}$ | lovely$_{(0.596)}$ | bad$_{(0.948)}$ | badly$_{(0.467)}$ | 295$_{(0.463)}$ |
| | 2048 | 0.0 | 1726$_{(0.328)}$ | ##た$_{(0.319)}$ | 1777$_{(0.309)}$ | 1726$_{(0.328)}$ | ##た$_{(0.319)}$ | 1777$_{(0.309)}$ |
| | | 0.2 | 1726$_{(0.354)}$ | ##た$_{(0.339)}$ | 1777$_{(0.338)}$ | 1726$_{(0.347)}$ | 1666$_{(0.330)}$ | ##た$_{(0.330)}$ |
| | | 0.4 | 1726$_{(0.386)}$ | 1777$_{(0.374)}$ | 1666$_{(0.373)}$ | bad$_{(0.383)}$ | 283$_{(0.377)}$ | 1744$_{(0.355)}$ |
| | | 0.6 | beautiful$_{(0.606)}$ | beautifully$_{(0.448)}$ | gorgeous$_{(0.445)}$ | bad$_{(0.648)}$ | 283$_{(0.377)}$ | 326$_{(0.374)}$ |
| | | 0.8 | beautiful$_{(0.884)}$ | gorgeous$_{(0.588)}$ | lovely$_{(0.536)}$ | bad$_{(0.908)}$ | badly$_{(0.418)}$ | good$_{(0.400)}$ |
| | 4096 | 0.0 | 1760$_{(0.483)}$ | 1709$_{(0.472)}$ | 1761$_{(0.472)}$ | 1760$_{(0.483)}$ | 1709$_{(0.472)}$ | 1761$_{(0.472)}$ |
| | | 0.2 | 1760$_{(0.518)}$ | 44th$_{(0.507)}$ | ##ذ$_{(0.502)}$ | 271$_{(0.503)}$ | 1760$_{(0.503)}$ | 1709$_{(0.503)}$ |
| | | 0.4 | beautiful$_{(0.605)}$ | 1760$_{(0.535)}$ | 1738$_{(0.528)}$ | bad$_{(0.612)}$ | 295$_{(0.518)}$ | 271$_{(0.514)}$ |
| | | 0.6 | beautiful$_{(0.825)}$ | gorgeous$_{(0.594)}$ | magnificent$_{(0.551)}$ | bad$_{(0.845)}$ | 295$_{(0.480)}$ | 321$_{(0.469)}$ |
| | | 0.8 | beautiful$_{(0.963)}$ | gorgeous$_{(0.635)}$ | lovely$_{(0.579)}$ | bad$_{(0.970)}$ | good$_{(0.442)}$ | badly$_{(0.432)}$ |
| 5e-05 | 1024 | 0.0 | ㅍ$_{(0.771)}$ | ㅋ$_{(0.758)}$ | 1738$_{(0.754)}$ | ㅍ$_{(0.771)}$ | ㅋ$_{(0.758)}$ | 1738$_{(0.754)}$ |
| | | 0.2 | ㅍ$_{(0.759)}$ | ㅋ$_{(0.754)}$ | 1738$_{(0.753)}$ | ㅍ$_{(0.743)}$ | 1738$_{(0.742)}$ | 1732$_{(0.738)}$ |
| | | 0.4 | beautiful$_{(0.776)}$ | 1738$_{(0.695)}$ | ㅋ$_{(0.694)}$ | bad$_{(0.768)}$ | 295$_{(0.661)}$ | 283$_{(0.654)}$ |
| | | 0.6 | beautiful$_{(0.917)}$ | gorgeous$_{(0.687)}$ | magnificent$_{(0.623)}$ | bad$_{(0.923)}$ | 295$_{(0.536)}$ | 283$_{(0.533)}$ |
| | | 0.8 | beautiful$_{(0.984)}$ | gorgeous$_{(0.663)}$ | lovely$_{(0.597)}$ | bad$_{(0.986)}$ | badly$_{(0.448)}$ | good$_{(0.446)}$ |
| | 2048 | 0.0 | 1729$_{(0.485)}$ | 1744$_{(0.484)}$ | 1756$_{(0.481)}$ | 1729$_{(0.485)}$ | 1744$_{(0.484)}$ | 1756$_{(0.481)}$ |
| | | 0.2 | 1729$_{(0.513)}$ | 1744$_{(0.510)}$ | 1734$_{(0.510)}$ | 1729$_{(0.512)}$ | 298$_{(0.505)}$ | 1732$_{(0.505)}$ |
| | | 0.4 | beautiful$_{(0.610)}$ | 1738$_{(0.531)}$ | 1734$_{(0.530)}$ | bad$_{(0.609)}$ | 298$_{(0.516)}$ | 1729$_{(0.513)}$ |
| | | 0.6 | beautiful$_{(0.819)}$ | gorgeous$_{(0.610)}$ | lovely$_{(0.550)}$ | bad$_{(0.836)}$ | 295$_{(0.479)}$ | 298$_{(0.476)}$ |
| | | 0.8 | beautiful$_{(0.959)}$ | gorgeous$_{(0.645)}$ | lovely$_{(0.580)}$ | bad$_{(0.967)}$ | good$_{(0.441)}$ | badly$_{(0.433)}$ |
| | 4096 | 0.0 | 1655$_{(0.417)}$ | 1717$_{(0.413)}$ | ##之$_{(0.412)}$ | 1655$_{(0.417)}$ | 1717$_{(0.413)}$ | ##之$_{(0.412)}$ |
| | | 0.2 | 1655$_{(0.448)}$ | 29th$_{(0.447)}$ | 1717$_{(0.442)}$ | 266$_{(0.444)}$ | 495$_{(0.439)}$ | 207$_{(0.437)}$ |
| | | 0.4 | beautiful$_{(0.556)}$ | 1738$_{(0.475)}$ | 29th$_{(0.474)}$ | bad$_{(0.567)}$ | 266$_{(0.462)}$ | 495$_{(0.462)}$ |
| | | 0.6 | beautiful$_{(0.787)}$ | gorgeous$_{(0.562)}$ | lovely$_{(0.514)}$ | bad$_{(0.811)}$ | 283$_{(0.450)}$ | 304$_{(0.444)}$ |
| | | 0.8 | beautiful$_{(0.952)}$ | gorgeous$_{(0.626)}$ | lovely$_{(0.568)}$ | bad$_{(0.961)}$ | good$_{(0.429)}$ | badly$_{(0.418)}$ |
| | | 1.0 | beautiful$_{(1.000)}$ | gorgeous$_{(0.619)}$ | lovely$_{(0.557)}$ | bad$_{(1.000)}$ | good$_{(0.450)}$ | badly$_{(0.401)}$ |

Table 7: The text and cosine similarity metrics of the three words decoded from RoBERTa-large, exhibiting the highest cosine similarity between the word embeddings in RoBERTa-large and the mixup embeddings using the nouns ‘characters’ and ‘doubts’

| Parameter | | Text | **characters** | | | **doubts** | | |
|---|---|---|---|---|---|---|---|---|
| $\mathrm{lr}_{VQ}$ | Codebook | $\lambda$ | Rank 1 | Rank 2 | Rank 3 | Rank 1 | Rank 2 | Rank 3 |
| 2e-05 | 1024 | 0.0 | 1736$_{(0.642)}$ | 1732$_{(0.630)}$ | 45th$_{(0.629)}$ | [unused659]$_{(0.595)}$ | [unused80]$_{(0.593)}$ | [unused176]$_{(0.592)}$ |
| | | 0.2 | 1736$_{(0.650)}$ | 1743$_{(0.641)}$ | 1732$_{(0.640)}$ | [unused659]$_{(0.636)}$ | [unused80]$_{(0.633)}$ | [unused176]$_{(0.632)}$ |
| | | 0.4 | 1736$_{(0.639)}$ | 1743$_{(0.636)}$ | 1732$_{(0.632)}$ | doubts$_{(0.703)}$ | [unused659]$_{(0.662)}$ | [unused276]$_{(0.662)}$ |
| | | 0.6 | characters$_{(0.812)}$ | protagonists$_{(0.605)}$ | 1743$_{(0.591)}$ | doubts$_{(0.863)}$ | [unused306]$_{(0.658)}$ | [unused298]$_{(0.658)}$ |
| | | 0.8 | characters$_{(0.952)}$ | character$_{(0.622)}$ | protagonists$_{(0.581)}$ | doubts$_{(0.968)}$ | doubted$_{(0.679)}$ | doubt$_{(0.615)}$ |
| | 2048 | 0.0 | 1726$_{(0.328)}$ | ##た $_{(0.319)}$ | 1777$_{(0.309)}$ | 1726$_{(0.328)}$ | ##た $_{(0.319)}$ | 1777$_{(0.309)}$ |
| | | 0.2 | 1726$_{(0.358)}$ | ##た $_{(0.338)}$ | 1744$_{(0.335)}$ | 1726$_{(0.378)}$ | ##た $_{(0.366)}$ | 1666$_{(0.360)}$ |
| | | 0.4 | characters$_{(0.472)}$ | 1726$_{(0.393)}$ | protagonist$_{(0.384)}$ | doubts$_{(0.525)}$ | 1726$_{(0.436)}$ | 1679$_{(0.423)}$ |
| | | 0.6 | characters$_{(0.696)}$ | protagonists$_{(0.470)}$ | protagonist$_{(0.456)}$ | doubts$_{(748)}$ | doubted$_{(0.522)}$ | [unused144]$_{(0.496)}$ |
| | | 0.8 | characters$_{(0.915)}$ | character$_{(0.595)}$ | protagonists$_{(0.527)}$ | doubts$_{(0.936)}$ | doubted$_{(0.631)}$ | doubt$_{(0.587)}$ |
| | 4096 | 0.0 | 1760$_{(0.483)}$ | 1709$_{(0.472)}$ | 1761$_{(0.472)}$ | [unused26]$_{(0.547)}$ | [unused609]]$_{(0.546)}$ | [unused757]$_{(0.546)}$ |
| | | 0.2 | 1760$_{(0.519)}$ | 44th$_{(0.508)}$ | 1764$_{(0.507)}$ | [unused26]$_{(0.598)}$ | [unused609]$_{(0.597)}$ | [unused248]$_{(0.596)}$ |
| | | 0.4 | characters$_{(0.626)}$ | 1760$_{(0.528)}$ | 1764$_{(0.527)}$ | doubts$_{(0.727)}$ | [unused440]$_{(0.630)}$ | [unused857]$_{(0.630)}$ |
| | | 0.6 | characters$_{(0.848)}$ | character$_{(0.579)}$ | protagonists$_{(0.563)}$ | doubts$_{(0.883)}$ | doubted$_{(0.643)}$ | [unused298]$_{(0.628)}$ |
| | | 0.8 | characters$_{(0.970)}$ | character$_{(0.642)}$ | protagonists$_{(0.550)}$ | doubts$_{(0.975)}$ | doubted$_{(0.670)}$ | doubt$_{(0.618)}$ |
| 5e-05 | 1024 | 0.0 | [unused467]$_{(0.840)}$ | [unused499]$_{(0.836)}$ | [unused962]$_{(0.836)}$ | [unused467]$_{(0.840)}$ | [unused499]$_{(0.836)}$ | [unused962]$_{(0.836)}$ |
| | | 0.2 | [unused467]$_{(0.807)}$ | [unused257]$_{(0.805)}$ | [unused962]$_{(0.804)}$ | [unused306]$_{(0.826)}$ | [unused962]$_{(0.826)}$ | [unused467]$_{(0.826)}$ |
| | | 0.4 | characters$_{(0.791)}$ | [unused257]$_{(0.708)}$ | [unused467]$_{(0.708)}$ | doubts$_{(0.876)}$ | [unused306]$_{(0.770)}$ | [unused313]$_{(0.768)}$ |
| | | 0.6 | characters$_{(0.925)}$ | protagonists$_{(0.638)}$ | protagonist$_{(0.596)}$ | doubts$_{(0.957)}$ | doubted$_{(0.703)}$ | [unused306]$_{(0.687)}$ |
| | | 0.8 | characters$_{(0.986)}$ | character$_{(0.621)}$ | Characters$_{(0.572)}$ | doubts$_{(0.992)}$ | doubted$_{(0.684)}$ | doubt$_{(0.631)}$ |
| | 2048 | 0.0 | 1729$_{(0.485)}$ | 1744$_{(0.484)}$ | 1756$_{(0.481)}$ | [unused158]$_{(0.573)}$ | [unused439]$_{(0.572)}$ | [unused306]$_{(0.572)}$ |
| | | 0.2 | 1744$_{(0.517)}$ | 1729$_{(0.516)}$ | 1734$_{(0.515)}$ | [unused306]$_{(0.626)}$ | [unused158]$_{(0.623)}$ | [unused439]$_{(0.622)}$ |
| | | 0.4 | characters$_{(0.633)}$ | 1734$_{(0.535)}$ | 1744$_{(0.529)}$ | doubts$_{(0.748)}$ | [unused306]$_{(0.657)}$ | [unused2]$_{(0.650)}$ |
| | | 0.6 | characters$_{(0.842)}$ | character$_{(0.569)}$ | protagonists$_{(0.559)}$ | doubts$_{(0.896)}$ | doubted$_{(0.647)}$ | [unused306]$_{(0.645)}$ |
| | | 0.8 | characters$_{(0.967)}$ | character$_{(0.637)}$ | protagonists$_{(0.550)}$ | doubts$_{(0.978)}$ | doubted$_{(0.670)}$ | doubt$_{(0.614)}$ |
| | 4096 | 0.0 | 1655$_{(0.417)}$ | 1717$_{(0.413)}$ | ##た $_{(0.412)}$ | [unused892]$_{(0.578)}$ | [unused949]$_{(0.574)}$ | [unused943]$_{(0.574)}$ |
| | | 0.2 | 1655$_{(0.456)}$ | ##た $_{(0.452)}$ | 266$_{(0.449)}$ | [unused892]$_{(0.628)}$ | [unused949]$_{(0.627)}$ | [unused943]$_{(0.627)}$ |
| | | 0.4 | characters$_{(0.577)}$ | ##成 $_{(0.483)}$ | ##志 $_{(0.707)}$ | doubts$_{(0.743)}$ | [unused298]$_{(0.716)}$ | [unused943]$_{(0.654)}$ |
| | | 0.6 | characters$_{(0.813)}$ | protagonists$_{(0.526)}$ | character$_{(0.510)}$ | doubts$_{(0.895)}$ | doubted$_{(0.658)}$ | [unused298]$_{(0.642)}$ |
| | | 0.8 | characters$_{(0.961)}$ | character$_{(0.614)}$ | protagonists$_{(0.539)}$ | doubts$_{(0.978)}$ | doubted$_{(0.675)}$ | doubt$_{(0.616)}$ |
| | | 1.0 | characters$_{(1.000)}$ | character$_{(0.647)}$ | protagonists$_{(0.502)}$ | doubts$_{(1.000)}$ | doubted$_{(0.657)}$ | doubt$_{(0.634)}$ |

ilarity indicates that those embeddings with the highest cosine similarity exhibit inferior semantic coherence compared to RoBERTa-large.

When $\lambda$ is set to 0, the mixup embeddings occupy a space distinct from the real embeddings. This phenomenon arises due to the non-convergence of the cosine similarity depicted in Figure 4c, despite the partial convergence of the reconstruction loss illustrated in Figure 4d. Furthermore, it is apparent that the mixup embeddings are predominantly characterized by synthetic embeddings when $\lambda$ is 0.2 and 0.4, with only embeddings being identified in a space similar to synthetic embeddings. For $\lambda$ values of 0.6 and 0.8, the mixup embeddings exhibit a greater resemblance to the real embeddings, with a minority of embeddings situated in a space similar to that of the synthetic embeddings. This observation substantiates that the mixup embeddings for BERT-large do not possess a semantic meaning that is markedly distinct from the real and synthetic embeddings.

Tables 4 and 5 illustrate that VQ-TEGAN generates semantic embeddings that provide RoBERTa-large with access to a more diverse and meaningful embedding space for learning. Conversely, Tables 6 and 7 reveal that the mixup embeddings on BERT-large exhibit less significant cosine similarity compared to those augmented on RoBERTa-large embeddings.